澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Reinforcement Learning

GongYing
gong.ying@connect.umac.mo
2023.12.18

Contents

- PPO

  PPO-penalty

  PPO-clip

- GAIL

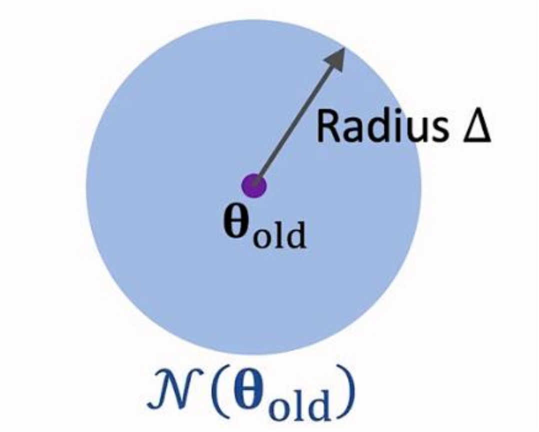
PPO

- PPO: Proximal Policy Optimization
- In TRPO, we use KL divergence to constrain the magnitude of the update.

Trust region methods, target: $\max_{\theta} J(\theta)$

- Trust region: $N(\theta_{now}) = \{\theta \mid \|\theta - \theta_{now}\|_2 \leq \Delta\}$.
- Construct function $L(\theta|\theta_{now})$, satisfying:

  $L(\theta|\theta_{now})$ is very close to $J(\theta)$, $\forall\theta \in N(\theta_{now})$
- $J(\theta)$ can be replaced by $L(\theta|\theta_{now})$ when $\theta \in N(\theta_{now})$



PPO

- PPO: Proximal Policy Optimization
- In TRPO, we use KL divergence to constrain the magnitude of the update.

Train

- **Approximate**
  a) Present policy network parameter is θ_{now}. Use $\pi(a|s;\theta_{now})$ to control the agent, record trajectories: $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_n, a_n, r_n$.
  b) For all t, calculate u_t.
  c) Approximate function: $\tilde{L}(\theta|\theta_{now}) = \frac{1}{n}\sum_{t=1}^{n}\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{now})}\cdot u_t$.
- **Maximize**

$$\theta_{new} = argmax_{\theta}\tilde{L}(\theta|\theta_{now}); \ \boxed{s.t. \|\theta - \theta_{now}\|_2 \leq \Delta.}$$ constraint

KL divergence $\boxed{\frac{1}{t}\sum_{i=1}^{t}\mathrm{KL}\Big[\pi(\,\cdot\,|\,s_i;\,\boldsymbol{\theta}_{\mathrm{now}})\,\Big\|\,\pi(\,\cdot\,|\,s_i;\,\boldsymbol{\theta})\Big] \leq \Delta}$

PPO

- PPO: Proximal Policy Optimization
- In TRPO, we use KL divergence to constrain the magnitude of the update.
- However, if we use gradient based optimization, it is difficult to deal with constraints.
- While in PPO, constraint is placed in the formula needed to be optimized:

$$\begin{cases} J_{PPO}(\theta|\theta') = J(\theta|\theta') - \beta \cdot KL(\theta, \theta') \\ \boxed{J(\theta|\theta') = E_{(s_t, a_t) \sim \pi(\theta')}\left[\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta')} A(s_t, a_t; \theta')\right]} \end{cases}$$

- KL divergence is now in target and it's easier to calculate.
- Note that KL divergence measures the similarity of probability distributions (actions) instead of parameters (θ and θ').

PPO

$$\begin{cases} J_{PPO}(\theta|\theta') = J(\theta|\theta') - \beta \cdot KL(\theta, \theta') \\ \boxed{J(\theta|\theta') = E_{(s_t,a_t)\sim\pi(\theta')}\left[\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta')} A(s_t, a_t; \theta')\right]} \end{cases}$$

In TRPO, $J(\theta) = E_S(V_\pi(S))$.

$$\because V_\pi(s) = \sum_{a\in A} \pi(a|s;\theta_{now}) \cdot \frac{\pi(a|s;\theta)}{\pi(a|s;\theta_{now})} \cdot Q_\pi(s,a)$$

$$= E_{A\sim\pi(\cdot|s;\theta_{now})}\left[\frac{\pi(A|s;\theta)}{\pi(A|s;\theta_{now})} \cdot Q_\pi(s,A)\right]$$

$$\therefore J(\theta) = E_S\left[E_{A\sim\pi(\cdot|S;\theta_{now})}\left[\frac{\pi(A|S;\theta)}{\pi(A|S;\theta_{now})} \cdot Q_\pi(S,A)\right]\right].$$

PPO - penalty

- **Target function**: $J_{PPO}(\theta|\theta^k) = J(\theta|\theta^k) - \beta \cdot KL(\theta, \theta^k)$
- In each iteration, use θ^k to interact with environment and sample (s_t, a_t), and then update θ.
- Here the problem is how to set value of β.
- **Adaptive KL divergence:** adjust β dynamically.

  Set a KL_{max} and a KL_{min}, if $KL(\theta, \theta^k)$> KL_{max}, it means $\beta \cdot KL(\theta, \theta^k)$ is too weak, then enlarge β (and vice versa).

$$\begin{cases} KL(\theta, \theta^k) > KL_{max}, & amplify\ \beta \\ KL(\theta, \theta^k) < KL_{max}, & diminish \beta \end{cases}$$

$$\begin{cases} J_{PPO}(\theta|\theta^k) = J(\theta|\theta^k) - \beta \cdot KL(\theta, \theta^k) \\ J(\theta|\theta') \approx \sum_{(s_t,a_t)} \dfrac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} A(s_t, a_t; \theta^k) \end{cases}$$

PPO - clip

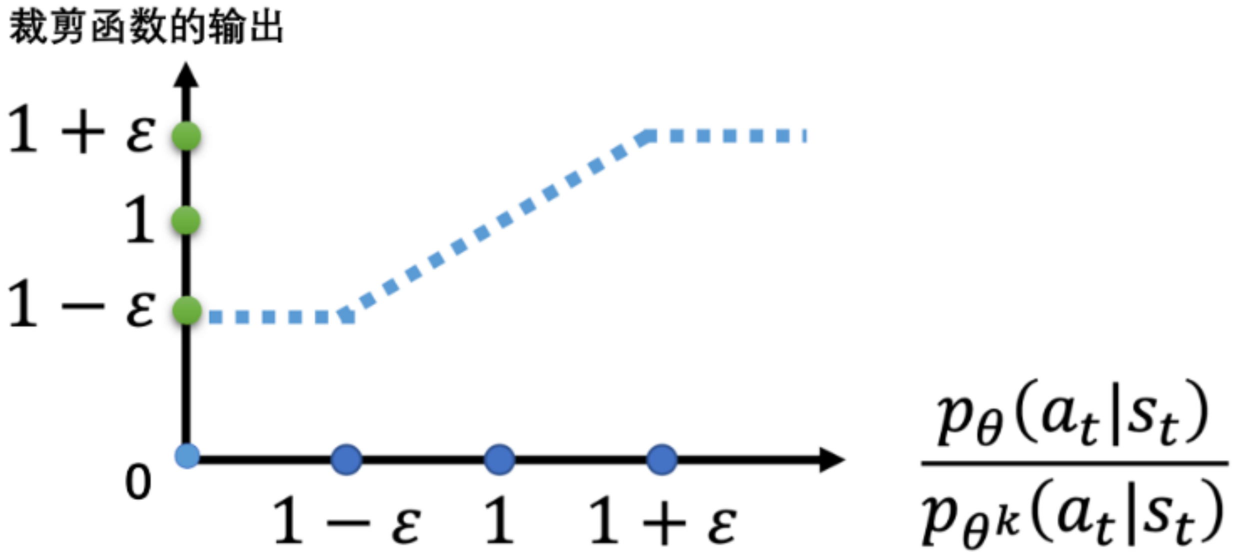
- In PPO2, clipping is introduced instead of KL divergence.
- **Target function**:

$$J_{PPO2}(\theta|\theta^k) \approx \sum_{(s_t,a_t)} \min\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} A(s_t,a_t;\theta^k),\ clip\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, 1-\varepsilon, 1+\varepsilon\right) A(s_t,a_t;\theta^k)\right)$$

- **clip()**: $clip\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, 1-\varepsilon, 1+\varepsilon\right) = \begin{cases} 1-\varepsilon, & \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} < 1-\varepsilon \\ 1+\varepsilon, & \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} > 1+\varepsilon \\ \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, & else \end{cases}$

PPO - clip

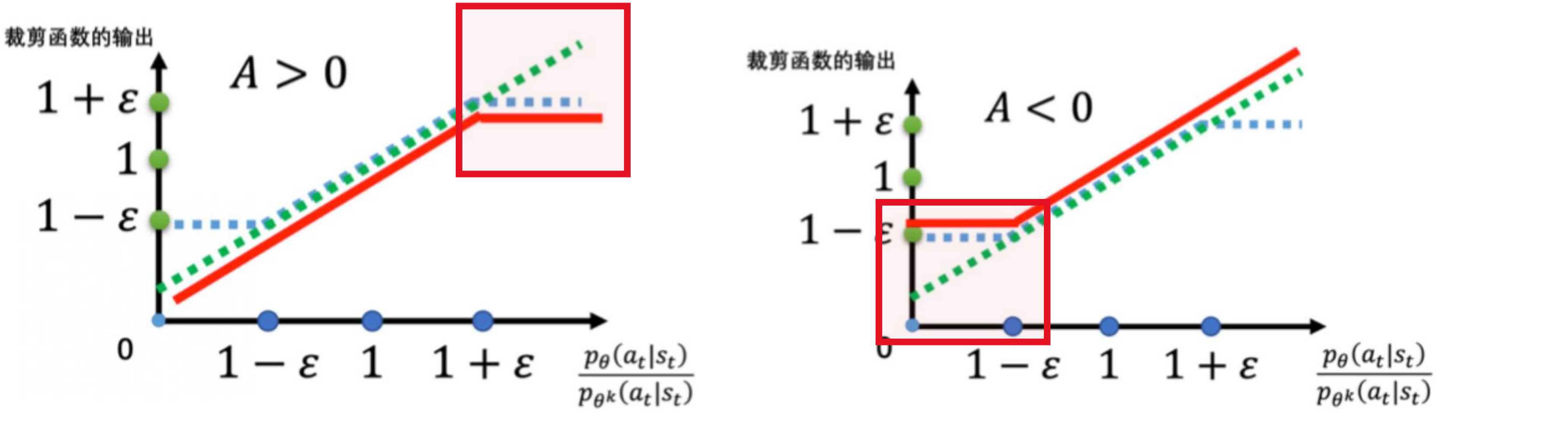
- **clip()**: $clip\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, 1-\varepsilon, 1+\varepsilon\right) = \begin{cases} 1-\varepsilon, & \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} < 1-\varepsilon \\ 1+\varepsilon, & \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)} > 1+\varepsilon \\ \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, & else \end{cases}$



PPO - clip

- **Target function**:

$$J_{PPO2}(\theta|\theta^k) \approx \sum_{(s_t,a_t)} \min\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}A(s_t,a_t;\theta^k), clip\left(\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta^k)}, 1-\varepsilon, 1+\varepsilon\right)A(s_t,a_t;\theta^k)\right)$$



(a) $A > 0$ (b) $A < 0$

In brief, $\pi(a_t|s_t;\theta)$ is supposed to be close to $\pi(a_t|s_t;\theta^k)$, at the same time, if A>0, ignore the advantage caused by over-deviation (and vice versa).

GAIL

- **GAIL** is based on **GAN** (generative adversarial network), which contains **generator** and **discriminator**.
  - **Generator** is used to generate fake samples.
  - **Discriminator** is used to determine wether a sample is fake or not.
- For example, generator generates a fake face picture, while discriminator determines whether it is generated by generator.
- In **GAIL**, data to be trained is the trajectories generated by human expert (imitated object):

$$\tau = [s_1, a_1, \dots, s_m, a_m].$$

- Data set contains k trajectories, denoted as:

$$\boldsymbol{X} = \{\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(k)}\}.$$

GAIL

- Trajectory:

$$\tau = [s_1, a_1, \dots, s_m, a_m].$$

- Data set contains k trajectories, denoted as:

$$\boldsymbol{X} = \{\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(k)}\}.$$

- **Generator**: $\pi(a|s; \boldsymbol{\theta})$,

  Input: s,

  Output: $\boldsymbol{f} = \pi(\cdot\,|s; \boldsymbol{\theta})$.

  $a_t \sim \pi(\cdot\,|s; \boldsymbol{\theta}),\ s_{t+1} \sim p(\cdot\,|s_t, a_t)$

GAIL

- Trajectory:

$$\tau = [s_1, a_1, \dots, s_m, a_m].$$

- Data set contains k trajectories, denoted as:

$$\boldsymbol{X} = \{\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(k)}\}.$$

- **Discriminator**: $D(s, a; \phi)$,

  Input: s,

  Output: $\widehat{\boldsymbol{p}} = D(s, \cdot \mid \phi)$, element $\widehat{p_a} = D(s, a; \phi) \in (0,1), \forall a \in \boldsymbol{A}$,

  1 means real (human expert) and 0 means fake (generator).

GAIL

- **Train**:
  1. Sample a trajectory from the training data set, denoted as:
  2. Use $\pi(a|s;\theta_{now})$ to control the agent, getting a trajectory, denoted as:
  3. Use Discriminator to determine if the actions of polity network is real:
  $$u_t = lnD\left(s_t^{fake}, a_t^{fake}; \phi_{now}\right), \forall t = 1, \dots, n.$$
  4. Take τ^{fake} and u_t as input, update the parameter of generator (policy network), getting θ_{new}:
  $$\theta_{new} = argmax_{\theta} \tilde{L}(\theta|\theta_{now}); s.t. dist(\theta_{now}, \theta) \leq \delta.$$
  5. Take τ^{real} and τ^{fake} as input, update the parameter of discriminator, getting ϕ_{new}:

$$\begin{cases} \phi \leftarrow \phi - \eta \cdot \nabla_{\phi} F(\tau^{real}, \tau^{fake}; \phi) \\ \boxed{F\left(\tau^{real}, \tau^{fake}; \phi\right)} = \boxed{\frac{1}{m}\sum_{t=1}^{m} \ln\left[1 - D\left(s_t^{real}, a_t^{real}; \phi\right)\right]} + \boxed{\frac{1}{n}\sum_{t=1}^{n} \ln D\left(s_t^{fake}, a_t^{fake}; \phi\right)} \end{cases}$$

Loss function

Smaller when D is larger

Smaller when D is smaller

.

Thank you.